
データベースの動的情報参照ツールの

開発について

富士通キャドテック株式会社

■ 執筆者 Profile ■



鈴木 悟 郎

- 1997年 富士通キャドテック株式会社入社
携帯電話修理管理システム開発に従事
- 1999年～ 富士通小山工場の生産管理システム
開発に従事
- 2001年～ 富士通通信部門のSCMシステム開発に
従事

■ 論文要旨 ■

富士通の基幹システムにおいて、大量なデータを扱う処理が頻繁に発生する中で、処理の完了時間、無限ループの有無、ボトルネックの存在等の情報が必要とされていた。

実際にはこのような情報はデータベース内※に存在し、効率よく利用することによりシステム開発者もしくは運用サポート等を行なう者にとって非常に有効な情報を提供することができる。

これらの情報を GUI により効率よく参照できるツールを開発した。

ツールの機能は下記の通りである。

1. データベース内の実行 SQL 文の検索
2. テーブルロック情報の検索
3. SQL 文の実行計画の参照やデータの抽出
4. セッション切断

本論文では、データベース内に存在する情報をいかに有効に、かつ効率よく提供できるか、という工夫と実際にツールを適用したことによる効果、そして今後の課題等について述べる。

※今回対象としたデータベースは Oracle 社製に限定する。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 背景	
2. 動的情報／実行計画とは	《 3》
2. 1 動的情報	
2. 2 実行計画	
3. 機能概要	《 5》
4. 画面構成および各機能について	《 6》
4. 1 動的情報参照画面	
4. 2 ロック情報参照画面	
4. 3 SQL実行画面	
5. ツール適用による効果	《 9》
5. 1 運用監視	
5. 2 システム開発	
6. 今後の課題と展望	《 10》
6. 1 拡張機能について	
6. 2 外販パッケージとしてのビジネス展開	
7. おわりに	《 11》

■ 図表一覧 ■

図1 動的情報参照ツールメイン画面	《 6》
図2 動的情報参照画面	《 7》
図3 SQL表示画面	《 7》
図4 実行計画表示画面	《 7》
図5 ロック情報参照画面	《 8》
図6 SQL実行画面	《 8》

1. はじめに

1. 1 背景

富士通の通信機器製造工場の生産管理システムという非常に大規模なクライアント／サーバシステムの開発担当者として、主にバッチ処理の開発を行ってきたが、その処理においては、生産管理だけでなく、部品倉庫管理や、製造工程管理などのシステムとのリアルタイム連携が必要とされる。

これまでに、データ移行処理などのテスト中に非常に大量なデータを扱うために時間のかかる処理が存在し、その原因追求や対処、処理完了時間の予測を行うのに、かなりの工数を費やした。

そのため以下のような要望が同じプロジェクト内の開発者や運用監視担当者からあがった。

- ① 大量データの処理を行なう際の、処理完了時間の予測やボトルネックとなる処理の特定
- ② ①で発見されたボトルネック処理のパフォーマンスを向上するためのチューニング情報の照会

②では特に、データベースが分散配置され、データベースリンクによりデータを参照するような処理の場合、そのデータを抽出するための最適な SQL を生成することによりパフォーマンスの向上が図れる。

しかし、前述のような機能を満たすツールはいくつか存在するものの、必要ではない機能が含まれるものが大半でかつ高価であるという点や、必要な機能自体の操作性が十分ではない点など課題があった。そこで、社内で独自にツール開発を行うことにより、ユーザ※の意見を織り込み、機能の向上／追加を可能とし、さらに社内に拡大することで、ツール購入費用の削減を図ることとした。

※ 以後、ユーザとは社内の運用管理者、システム開発者を指す。

2. 動的情報／実行計画とは

本論文内で使用する動的情報、実行計画という用語について説明する。

※Oracle8 Sever リリース 8.0 リファレンスマニュアル抜粋

2. 1 動的情報

データベース内にはデータベース管理者がアクセスできる（ユーザが異なっても権限により他のユーザもアクセス可能）ビューが存在する。ビューとは参照のみ可能な表であり、データベースがオープンされている間は継続的に更新されるため、動的性能ビューと呼ばれる。（本論文内では動的情報と表記する）

その中には現在処理中のセッション、テーブルのロック情報や、オープン中の SQL など、さまざまな情報が存在し、キー項目によってそれぞれのビューがつながりを持っている。

動的情報を参照する際に必要となる各ビュー表の性質を以下に示す.

- ① オープンカーソル (V\$OPEN_CURSOR)
各ユーザ・セッションが現在オープンし解析しているカーソルの情報
オープン中の SQL 文字列の情報を保持
- ② セッション (V\$SESSION)
各ユーザ・セッション情報
セッションに関連する ID や状態, ユーザ, OS 端末などの情報を保持
- ③ 共有 SQL 領域 (V\$SQLAREA)
共有 SQL 領域についての統計情報
SQL 文やその SQL 文の実行回数, 処理対象行の合計数などの情報を保持
- ④ SQL テキスト (V\$SQLTEXT)
共有 SQL カーソルに属するオープンカーソルでは保持していない SQL 全文の情報を保持
- ⑤ ロック (V\$LOCK)
現在保持されているロック情報
ロックされているオブジェクトを示す識別子やそのロック状態などの情報を保持

2. 2 実行計画

実行計画とは, データベースが SQL を実行するために行う一連の処理で, その情報を参照するとにより, その実行方法がわかる. テーブルの結合方法や, 使用しているインデックスの情報を照会することができ, それを参照することにより, ある程度チューニングのためのヒントを得ることができる.

3. 機能概要

今回開発したツールは主にデータベース内の動的情報を、実行中の SQL のワイルドカードでの検索や、ユーザ、セッション ID などの検索を行うことで、ユーザの见たい情報のみ表示できることを目的として作成した。さらに同一画面からのセッション切断やサブ画面での補助情報の表示など、ユーザの要望を折り込んだものとした。

① 動的情報照会画面

当ツールのメイン画面であり、現在実行中の SQL の状態を、実行中の SQL のワイルドカードでの検索や、ユーザ、セッション ID での検索など、さまざまな条件により参照でき、さらに該当レコードのセッション切断や、SQL の全文表示などが行える。

② ロック情報参照画面

現在ロックされているテーブルの状態やプロセス情報を表示し、該当レコードのセッション切断も可能とした。

③ SQL 実行画面

SQL (Select 文のみ) をユーザが発行し、その結果を表示する。また、その SQL の実行計画を参照できる。

※動的情報参照の機能とは異なるが、パフォーマンス向上のための機能としてユーザ要望により追加した。

④ SQL スキャン画面

オブジェクト (パッケージ、ストアドファンクション、ストアドプロシージャ) 内の SQL を全て抽出し照会できる。ただし、この機能は現在 SQL を参照するだけだが、将来的にはその実行計画を取得できるように拡張する予定である。

(この機能については後述する)

4. 画面構成および各機能について

当ツールはまずマルチドキュメントインターフェース形式のメイン画面（**図 1**）からなり、必要に応じて各サブ画面を起動することとなる。

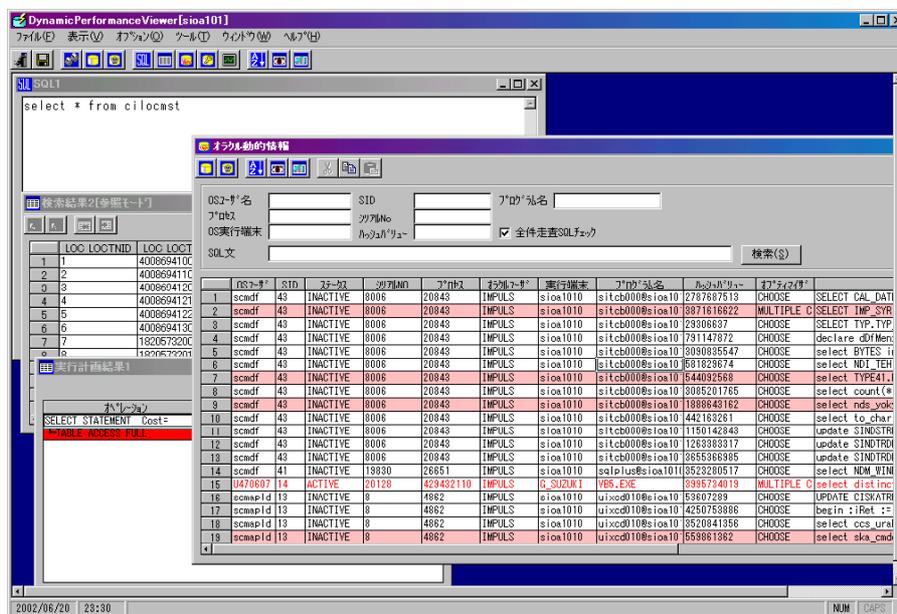


図 1 動的情報参照ツールメイン画面

4. 1 動的情報照会画面

図 2に画面レイアウトを示す。

当画面は、前述の動的パフォーマンスビューのうち、V\$OPEN_CURSOR, V\$SESSION, V\$SQLAREA, V\$SQLTEXT という4つの表を結合し、ある処理の状況を把握するための情報を集約して表示する。

特徴と工夫点

今回開発したツールの特徴および使いやすくなるための工夫点は下記の通りである。

① 検索条件指定

従来のツールでは存在するセッション情報を全て表示していたが、ユーザ名やプロセス、SQLのワイルドカード検索を可能とした。

これによりユーザは自分が現在実行しているセッション、もしくはあるテーブルに対し、アクセスしているセッションのみを参照するといった絞込みが可能となった。

② SQL全文表示

動的情報画面のメイン画面では、一覧表示のSQLは一部しか参照できない。そこで、そのセルをダブルクリックすることでサブ画面（**図 3**）が起動し、SQLの全文が表示される。

③ SQL 実行計画参照

②で表示されたサブ画面からさらに[実行計画]ボタンを押下することで、実行計画結果のサブ画面 (図 4) が起動し、その SQL の解析結果を見ることが可能となる。

これにより非効率なテーブル結合をしている SQL の判断や、改善するためのヒントを提供することが可能となる。

④ 全件走査 SQL 表示

検索時に[全件走査 SQL チェック]をチェックして検索することで、SQL が全件走査を行っている行の背景色を変更し、一目で処理の遅延の原因となっている SQL を探し出すことが可能となる。

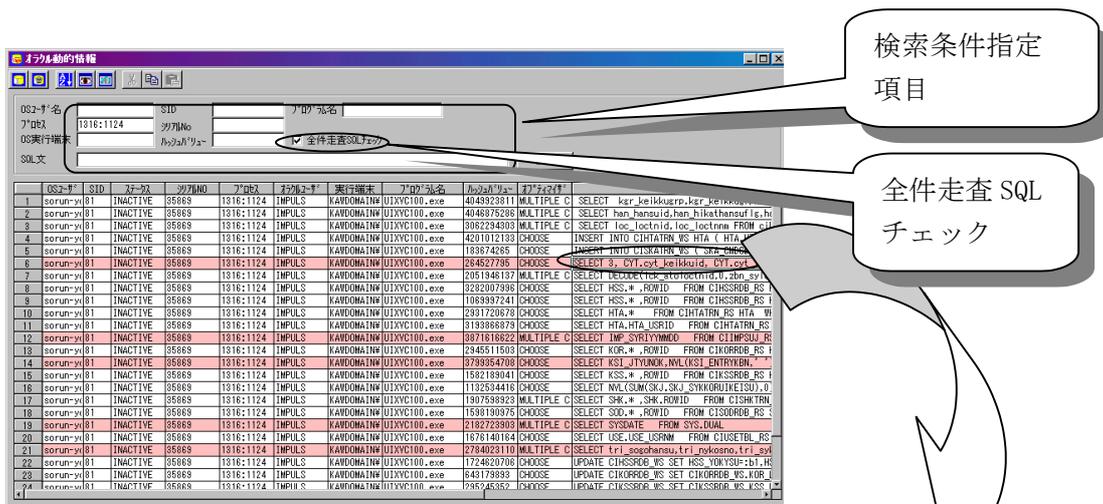


図 2 動的情報参照画面

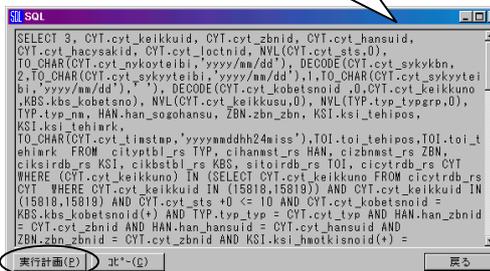


図 3 SQL 表示画面

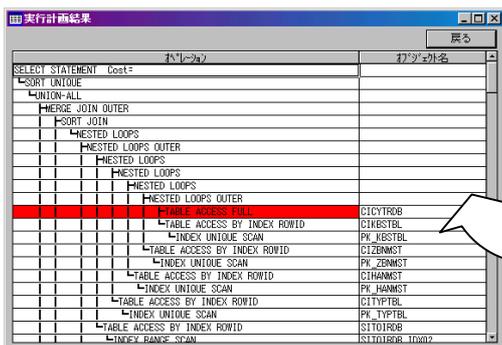


図 4 実行計画表示画面

4. 2 ロック情報照会画面

図5に画面レイアウトを示す。

当画面では、前述の動的表のうち、V\$LOCK, V\$SESSION, OBJ\$という3つの表を結合し、現在ロックされているテーブルの情報にさらにセッション情報を付与して表示する。

オブジェクト名	ロックタイプ	ロックモード	経過時間(秒)	OSユーザ	SID	状態	オブジェクトNo	アドレス	行番号	
1	行ロック	排他	0	scott	23	ACTIVE	80144	17467	IMPULS	
2	CIHTATRN	表ロック	行共有	0	scott	23	ACTIVE	80144	17467	IMPULS
3		行ロック	排他	839	A00008	50	INACTIVE	22561	429440146	IMPULS
4	CIKSSRDB	表ロック	行排他	806	A00008	50	INACTIVE	22561	429440146	IMPULS
5	CIKZSRDB	表ロック	行排他	806	A00008	50	INACTIVE	22561	429440146	IMPULS
6	CIUMGDLJ	表ロック	行排他	836	A00008	50	INACTIVE	22561	429440146	IMPULS
7	CIHKTRN	表ロック	行排他	809	A00008	50	INACTIVE	22561	429440146	IMPULS
8	CIKJRDB	表ロック	行排他	793	A00008	50	INACTIVE	22561	429440146	IMPULS
9	CIOSHTRN	表ロック	行排他	809	A00008	50	INACTIVE	22561	429440146	IMPULS
10	CIHSSRDB	表ロック	行排他	806	A00008	50	INACTIVE	22561	429440146	IMPULS
11	CICTSTRN	表ロック	行排他	803	A00008	50	INACTIVE	22561	429440146	IMPULS
12	CIKORDB	表ロック	行排他	806	A00008	50	INACTIVE	22561	429440146	IMPULS
13	CIOTWRK	表ロック	行排他	786	A00008	50	INACTIVE	22561	429440146	IMPULS
14	CIKSTRN	表ロック	行排他	809	A00008	50	INACTIVE	22561	429440146	IMPULS
15	CIKWRK	表ロック	行排他	839	A00008	50	INACTIVE	22561	429440146	IMPULS
16	CIKORDB	表ロック	行排他	806	A00008	50	INACTIVE	22561	429440146	IMPULS
17	CIOSRDB	表ロック	行排他	800	A00008	50	INACTIVE	22561	429440146	IMPULS
18	CIKORDB	表ロック	行排他	806	A00008	50	INACTIVE	22561	429440146	IMPULS
19	CICTYRDB	表ロック	行排他	803	A00008	50	INACTIVE	22561	429440146	IMPULS
20	CIOSMPTR	表ロック	行排他	793	A00008	50	INACTIVE	22561	429440146	IMPULS

図5 ロック情報参照画面

4. 3 SQL実行画面

図6にSQL実行画面を示す。

当画面からSQLを実行された結果を表示する画面と、実行計画の結果を出力するサブ画面に遷移する。動的情報照会画面と同様に実行計画を参照することができる。処理で実際に発行するSQLをこの画面から発行し、事前にある程度のチューニングを行うことができる。

SQL実行画面のスクリーンショット。SQL文の入力欄には「SELECT * FROM SYOHIN」が入力されている。実行ボタン「実行計画(P)」が強調されている。実行計画の結果は「TABLE ACCESS FULL」を示している。実行結果の表示は以下の通りである。

B CODE	S CODE	S NAME	TANKA	SHI T	
1	00001	11001	ウイスキー	3600	2700
2	00001	11002	焼酎	1030	600
3	00001	11003	酒	1800	1350
4	00002	12001	コーラ	106	80
5	00002	12002	ホカリスエット	106	80
6	00002	12003	ジャルトコーラ	106	80
7	00003	13001	オレンジジュース	106	80
8	00003	13002	グレープジュース	106	80
9	00003	13003	パインジュース	106	80
10	00004	14001	ヤクルト	70	50
11	00004	14002	35牛乳	198	150
12	00004	14003	コービー牛乳	250	180
13	00005	15001	レモン	198	120
14	00005	15002	リンゴ	550	380
15	00005	15003	ブドウ	900	650
16	00006	16001	大根	198	110
17	00006	16002	キャベツ	250	150
18	00006	16003	レタス	300	180
19	00007	17001	こしむか	1180	980
20	00007	17002	特選コシ	1180	980

図6 SQL実行画面

5. ツール適用による効果

本ツールを適用したことにより、2つの用途において以下のような効果が得られた。

5.1 運用監視

今まで処理の実行状況を把握するには、本ツールで発行しているような SQL をプロンプトや、既存のツールから発行して、その結果をみて判断していた。その場合、処理対象のデータのみを抽出することや、その進度を迅速に把握するには限界があった。毎回全セッションデータを抽出して、その中から自分が参照したいセッションを見つけ出し、さらにその条件で抽出する SQL を変更して再発行するといった手間のかかる作業を行う必要があったため非常に作業工数がかかっていた。

また遅延している処理が存在した場合にその原因となる SQL を探し出すのはさらに手間のかかる作業であった。

本ツールではそういったことを意識せずに、条件を指定するだけで自分がみたいセッションやユーザの最新の情報を効率よく抽出することが可能となった。

これにより、一連の手間のかかる作業を省くことができ約 30%の工数削減となった。

5.2 システム開発

システムテストを行う際に処理性能の向上が課題となるが、実際には理論上最適 SQL を組み込んだと思っても、処理を実行した時に思ったほどのパフォーマンスがでないなどの差が生じる場合がある。今までは、ボトルネックとなる SQL を特定するには処理内の SQL すべてについてトレースをとり、その結果を見て判断していた。膨大なステップ数の処理においてそれを実行するには、多大な工数を費やしていた。

今回の動的情報参照画面ではそのボトルネックとなる SQL を全件走査 SQL の検索によりある程度の目星をつけることが可能になり、しかも対象の SQL について画面からの簡単な操作だけで実行計画の結果を見やすい形で参照できることになった。

今回、合計 5 つのパッケージで 10K ステップ程度の機能について、その有効性を検証した。

全体でボトルネックの対象となる SQL が 4 つ存在し、それらを従来の方法と、今回開発したツールを使用した場合で、パフォーマンスを向上させるために費やした工数を比較すると、従来の方法では 5 日費やしたものがツールを使用することで 2 日で同等のパフォーマンスを実現することが可能となった。

また、上記のような効果はいくつかの市販ツールを組み合わせることで実現可能ではあるが、市販ツールの購入費がかかることを考慮すると、今回の開発により金額面においてもかなりの効果が得られている。

6. 今後の課題と展望

6. 1 拡張機能について

背景においても述べたが、社内で独自に開発しているツールであるがゆえに本ツールの機能を拡張していくことは、容易である。

その一つとして現在、データベース内のプロシージャやファンクションについて、内部で発行している SQL を抽出する機能を開発している。

プロシージャ内の SQL は、通常ツールからは発行できないような形式になっていることが多く、また余計なコメントも含まれているが、それらを編集し表示する。ここまでは現在でも実現しているが、このままでは処理内部の変数や定数が含まれた形式で表示されてしまうため、SQL を発行できない。この定数をなんらかの方法で変換し、SQL を発行することができれば、それにより対象となる SQL の実行計画を取得することができ、結果としてプロシージャなどを実行する前に非効率なテーブル結合を行っている SQL の洗い出しが行えるようになる。

これは非常に重要で、例えば単体テストレベルでは処理時間が短く、動的情報参照画面で実行中の SQL 情報を抜き出すことが不可能な場合や、対象の SQL を発行する処理を経由しない場合でもその実行計画結果を取得することが可能となる。その結果、プログラム開発をする上で非常に重要であるパフォーマンスを実行前に改善できることとなり、開発効率の向上につながると思われる。

また、今回は動的情報参照という機能についてのツールとして開発したが、要望についてユーザにアンケート調査を行ったところ、SQL のテンプレート化や、データベース内部オブジェクト情報の参照といった機能の要望があげられた。

社内において結果的に効率化につながる機能であるという面や、後述するビジネス展開といったことを考慮すると、こうした要望も視野にいれて機能拡張を行っていく必要がある。

6. 2 外販パッケージとしてのビジネス展開

本ツールは業務効率化という目的で作成したものであるが、本ツールが持つ機能はシステム開発者全体にとっても非常に有効なものである。

そこで正式に外販パッケージ製品として作成するためのプロジェクトを組み、一般企業向けにビジネス展開を行い、会社全体としての利益拡大を図りたい。

実際に実現するためにはマーケティングリサーチやパッケージ化するための技術修得などを行うことが必要不可欠になるが、その見返りは大きい。今後そういったパッケージ製品化という方向も視野にいれ、業務効率化とともに並行して推進していきたい。

7. おわりに

今回ツールを開発することになるきっかけになったのは、実際にシステム開発を行っている人や、私自身が今回の開発したような機能を必要だと思ったからである。

普段、周りの人々から業務遂行上の非効率的な作業に対する意見や改善要望があったとしても、なんらかの解決案が実現されることはほとんどなかった。今後も常にこうした意見、要望に対し、どのように解決すればいいか、どんな機能があれば効率化につながるかなどのアイデアをなんらかの形で実現していきたいと思う。

また、アイデアを蓄積していけば、特許といった形で実現できる可能性もあるので、機会があれば積極的にチャレンジしていきたい。

最後に、今回のツールの開発にあたり、周りのユーザの方々にたくさんの意見や要望をいただいたことに感謝の意を表して、まとめとする。